STACKED NETWORK SWITCH USING RESILIENT PACKET RING COMMUNICATION PROTOCOL

Inventors: Brian H. W. Yang, Ken Ho, Aamer Latif

REFERENCE TO RELATED APPLICATIONS

[0001] This application claims priority to U.S. Prov. No. 60/463,992 filed April 18, 2003, incorporated herein by reference.

FIELD

[0002] The present invention relates to the field of telecommunications, and more particularly to a stacked network switch using resilient packet ring communication protocol.

BACKGROUND

[0003] Digital broadband networking and communications products and services are the infrastructure over which the Internet operates. The universal benefits of the Internet are well known, enabling immediate worldwide sharing of news and events, access to in-depth research on virtually any topic, sophisticated financial analysis available to all, the convenience of e-commerce available on virtually any product to consumers and the emerging capabilities for commercial e-commerce, and the outsourcing enabled by Application Service Providers and Storage Area Networks, to list just a few of the world-changing available uses.

[0004] This explosive growth in network traffic is further demonstrated by forecasts made by many leading networking industry experts regarding scaling specific infrastructure areas. Every aspect of these scaling estimates represents requirements for network equipment to scale to provide the necessary bandwidth.

[0005] Telecommunications switches help to meet the needs of many devices to connect to a network and then for the network to communicate with other networks. However, often there is a need for many ports (e.g. 128), which can exceed the number of ports in a standard switch (e.g. 32). In these cases, network engineers typically construct a stacked switch consisting of many interconnected switches. The simplest stacked switch simply connects an available port in one switch with an available port in another switch and utilizes a standard protocol between the two in

order to route the telecommunications traffic. A problem with this simple implementation is that the interconnected ports are no faster than the other ports (e.g. 10/100).

[0006] One improved technique of creating a stacked switch provides a proprietary high-speed interconnect between switches. This technique is an improvement because it provides for much faster traffic between the switches. However, a proprietary protocol does not support flexibility of stacked switch design and construction. It also may not support fault tolerance or other advanced features that would be desirable in a stacked switch.

[0007] What is needed is a stacked switch that uses a high-speed open standard communication protocol between the switches, and which has the ability to provide advanced features such as fault tolerance and communication port handover.

SUMMARY OF INVENTION

[0008] A stacked switch using a resilient packet ring protocol comprises a plurality of switch modules coupled to one another in a ring topology and each having a plurality of external terminals for interfacing with external devices. Each switch module includes an external interface for communicating with the external terminals, the external interface configured to communicate using a communication protocol (e.g. Ethernet protocol); and an internal interface for communicating with other switches, the internal interface using a resilient packet ring (RPR) protocol.

[0009] In one embodiment, each switch module further includes a controller coupled to the external interface and the internal interface and configured to selectively communicate information between the external interface and the internal interface.

[0010] In another embodiment, the stacked switch further comprises (a) a master management processor coupled to one or more switch modules and configured to provide instructions regarding the communication of information between each switches' external interface and internal interface, and to control data flow; and (b) a slave management processor coupled to the master management processor through at least one switch and one or more switch modules and configured to provide instructions regarding the communication of information between each switches' external interface and internal interface, and to control data flow. In one aspect of the invention, the processors assign their master/slave relationships based on some predetermined criteria and can re-assign the relationships based on fault conditions.

[0011] In one aspect of the invention, the stacked switch further comprises a link aggregation port coupled to at least two switch modules' external terminals and configured to selectively aggregate information to and from the switch modules.

[0012] Advantages of the invention include the ability to flexibly create a high performance stacked switch with advanced features.

BRIEF DESCRIPTION OF THE FIGURES

[0013] The present invention is illustrated by way of example, and not by way of limitation, in the figures of the accompanying drawings and in which like reference numerals refer to similar elements and in which:

[0014] Figure 1 depicts a ring-type and star-type stacked switch architectures for coupling the switch modules;

[0015] Figure 2 depicts a stacked switch using RPR according to an embodiment of the invention;

[0016] Figure 3 depicts a detailed view of Figure 2 showing the internal components of a stacked switch module according to an embodiment of the invention;

[0017] Figure 4 depicts link aggregation trunking over an RPR stack according to an embodiment of the invention;

[0018] Figure 5 depicts an exemplary persistent flooding problem in link aggregation across multiple modules;

[0019] Figure 6 depicts an exemplary conversation handover from one port to another port according to an embodiment of the invention;

[0020] Figure 7 depicts a technique for sending a marker to facilitate handover from one port to another port according to an embodiment of the invention;

[0021] Figure 8 depicts a technique for load balancing in a LAG configuration according to an embodiment of the invention; and

[0022] Figure 9 depicts a procedure for sending a marker PDU frame according to an embodiment of the invention.

DETAILED DESCRIPTION

[0023] The invention is described with reference to specific architectures and protocols. Those skilled in the art will recognize that the description is for illustration and to provide the best mode of practicing the invention. The description is not meant to be limiting. For example, reference is made to Ethernet Protocol and Resilient Packet Ring (RPR) Protocol but other protocols can be used in the invention.

[0024] Glossary

Stack: a set of switch chips (modules) connected by stacking links (e.g. RPR ring)

Module: a switch chip

Management processor: a processor handling the management layer function for a group of one or many modules; there are multiple management processors in a stack, one of them is master, others slaves

Routing processor: a processor handling the L3 routing and forwarding function for a group of one or many modules; there are multiple routing processors in a stack, one of them is master, others slaves

LAG: Link Aggregation Group

MMP: Master Management Processor

SMP: Slave Management Processor

MRP: Master Routing Processor

SRP: Slave Routing Processor

RPR: Resilient Packet Ring

10GE: 10 Gigabit Ethernet

[0025] A. Architecture

[0026] A definition of stacking in the context of the invention is to couple multiple individual switches together as a group to create a combination switch. In this context, a group of modules 110A-11D can be coupled, for example, though an RPR ring in a ring configuration or 10GE links in a star configuration.

[0027] Figure 1 depicts a ring-type and star-type stacked switch architectures comprising a number of modules 110A-110D, for example. The preferred architecture for the invention is a ring-type architecture, but other architectures can be implemented. The stacked switch is constructed from a number of switch modules (switch modules are labeled as Alpine) that are linked to one another and appear as a single L2 or L3 (level 2 or level 3) switch. The connection between the switches is via 10GE links or RPR ring, but the group is managed as a single switch. In the case that the switch appears as a since L3 switch, it may still have multiple Internet Protocol (IP) addresses. The stacked switch supports link aggregation among ports that belong to the same or different modules in the stack. The invention also supports mirroring among ports belong to same or different modules in the stack.

[0028] The ring architecture uses RPR as follows. The ring employs the resiliency mechanism defined in the RPR standard. Consequently, the invention doesn't need extra switch/fabric chips for stacking. However, in some cases, the ring can have a scalability issue since the ring may become a bottle neck (RPR ring only provide total 20G duplex bandwidth share by all Alpines, no matter how many Alpines in a stack).

[0029] The star architecture uses a 10 Gigabit Ethernet connection. In this case, the resiliency is achieve by redundant connections, and is less sophisticated than RPR. The switch may need extra switch/fabric chips for stacking connections. However, this architecture may scale better depending on the application and throughput requirements.

[0030] In the L2 scheme, there is no visibility of ports in other modules, L2 learning base on srcPortID (of own module) just like non-stacking case; both Ingress Alpine and egress Alpine need to perform L2 look up. (Adv: No special case, stacking traffic or not. No special encapsulation on stacking traffic. Disadv: high bandwidth requirement on L2 lookup/forwarding which is required anyway since need to support stand-alone 10GE interface)

[0031] B. Stacked Switch Ring

[0032] Figures 2 and 3 depict a stacked switch using RPR according to an embodiment of the invention. The modules 110A-110D are coupled to one another with a dual RPR ring, where a first ring employs clockwise communication and a second ring employs counter-clockwise communication. This dual-ring architecture improves performance and fault tolerance. Figure 3 depicts a detailed view of the switch modules including the internal RPR MAC (media access controller) and other components that facilitate the management and switching functions of the invention.

[0033] A management processor is coupled to each of the modules as a management controller. The processor controls functions such as various management tasks and routing processes. The management processor handles the management layer function for a group of one or many modules. In a stacked switch, there are multiple management processors, where one of them is the master (MMP) nd the others are slaves (SMP). A routing processor is one that handles the L3 routing and forwarding function for a group of one or many modules. In a stacked switch, there are multiple routing processors, where one of them is the master (MRP) and the others are slaves (SRP).

[0034] Figure 4 depicts link aggregation trunking over an RPR stack according to an embodiment of the invention. The Link aggregation is designed to support an external device that coupled to one or more stacked switch port. In one aspect of the invention, local ports on a switch are aggregated. In another aspect of the invention, ports on different switched are aggregated and an

external device couples to one port on one module and another port on another module. The invention has the management function to handle an aggregated link across modules.

[0035] Figure 5 depicts an exemplary persistent flooding problem in link aggregation across multiple modules. There is persistent flooding for all packets from HostB 130B to HostA 130A since reverse traffic never goes through module 110B, preventing it from properly learning HostA's MAC address. An example of persistent flooding problem can occur across multiple modules. The following is an example.

1. HostB with MAC==B send a packet, PKT0 (srcMAC=B, destMAC=A) to HostA with MAC==B; the interconnection path between HostA and HostB traverse across a pair of aggregated links in a Stack.
2. PKT0 enter module1 through West link of LAG1; assume destMAC==A had never been learnt in module1 before, so PKT0 will be flood to all ports include LAG0 West link to eventually reach HostA.
3. PKT0 reached HostA.
4. HostA generate a reverse directed packet, PKT1 (srcMAC=A, destMAC=B) back to HostB.
5. PKT1 enter module0 through East link of LAG0; if destMAC==B had been learnt by module0 before then PKT1 will not be flood but forwarded to LAG1 East link to reach HostB.
6. PKT1 reach HostB.

[0036] Since PKT1 was not being flooded to module1, the MAC address A will never be learnt by module1. Subsequent traffic from HostB to HostA will persistently be flooded by module1. To solve this problem, when module0 learn MAC A from an ingress packet entering a Link Aggregatioin Port, it communicate this learning to other modules so that they can be forced to learn that MAC address as well. This is achieved by software initiated intra-stack management frames.

[0037] Figure 6 depicts an exemplary conversation handover from one port to another port in a Link Aggregation Group (LAG) configuration according to an embodiment of the invention. Figure 7 depicts a technique for sending a marker to facilitate handover from one port to another port according to an embodiment of the invention.

[0038] The invention employs a marker technique for preventing an out-of-order problem when handing over traffic from one port to another. To prevent an out-of-order problem, the link aggregation standard requires that the same conversation, for example, (3-tuple: {srcMAC, destMAC, QOS}) must be consistently transmitted through the same port/link in a LAG. During

update of LAG configuration, one conversation can be switch from one physical port to the other within the same LAG. However, if there is conversation traffic during the re-configuration, an out-of-order condition may occur if not handled properly. The invention employs a marker frame scheme is used to solve this problem. There are two favors of the marker frame scheme (IEEE Scheme for single-module LAG and RMI extension for multi-module LAG).

1. Assume a particular LAG A with members {portA0, portA1, ..., portAn}.
2. Assume to move conversation bucket B from portAj to portAk.
3. Master processor command all slaves to update LAG A table in all modules to discard further incoming conversation bucket B packets.
4. Start timer (for timeout).
5. Send a marker PDU frame to output queue of portAj. (which should trigger the other end of the link aggregation link to response with a marker response PDU frame). Since each output queues consists of 8 priorities, we need a special procedure to send marker PDU frame.
6. Wait for either marker response PDU from portAj or time-out timer expire (this ensure all the conversation B traffic had been received by the other end).
7. Master processor command all slaves to update LAG A table in all modules to map conversation bucket B to portAk. (so that subsequent conversation B traffic will be transmitted to portAk) and stop discard of conversation bucket B traffic.
8. Conversation B traffic start transmit from portAk.

[0039] An exemplary LAG handover to move a conversation B from portA1 to portA2 is shown in Figure 7 by following the numbered arrows as follows.

1. MMP sends a request to SMPs to send a marker PDU frame to portA1, to discard incoming conversation B traffic.
2. SMP0 sends a marker PDU to portA1, SMPs change the LAG map at modules 0, 2, 4, and 5 to discard conversation B traffic.
3. Module 2 sends out a marker PDU frame to remote switch through portA1.
4. Module 2 receives a marker response PDU frame from portA1 and notifies SMP0.
5. SMP0 notifes MMP about reception of the marker response PDU.
6. MMP notifies SMPs to update the LAG map in all modules to transmit conversation to portA2.
7. The conversation handover is complete.

[0040] Figure 8 depicts a technique for load balancing among LAG links in a LAG configuration according to an embodiment of the invention. The following is an example of how to perform this function.

1. Traffic destined to a LAG is analyzed and then dynamically mapped (hashed) into conversation buckets (e.g. 32) from n-tuple, for example 3-tuple ({destMAC, srcMAC, priority}). The value of n and the form of information from the packet header depends on available space on an exemplary integrated circuit.
2. Each conversation bucket is then mapped into a number of physical output ports (e.g. 1 of up to 8) by LAG membership table.
3. Load balance is achieved by programming the LAG table in such a way that the among traffic (of one or many conversation buckets) to each port are more or less balanced.
4. In one aspect, the invention provides statistics based on LAG port on a per conversation bucket per port basis so that the software knows how much traffic a conversation bucket contains and can dynamically assign/move conversation buckets based on packet characteristics.
5. In one aspect, load balancing is preferrably on a conversation bucket granularity (more number of conversation bucket, finer the granularity), it is possible that majority of the traffic may belong to a single conversation bucket and thus prevent the conversations from being properly load balanced without modifying the hashing algorithm.
6. In one aspect, the invention includes synchronization.
7. In one aspect, a marker is selectively added to the data stream to provide a guide for the switching.

[0041] As described above, the invention can provide statistics based on LAG port traffic. The processors 112A-112D can use this information to selectively allocate port resources and create or delete link aggregation configurations. Also, as described above, the statistics can be used for load balancing purposes to aid the processors in their algorithmic decisions to selectively allocate port resources and create or delete link aggregation configurations.

[0042] Figure 9 depicts a procedure for sending a marker PDU frame according to an embodiment of the invention. The following is an example of how to perform this function.

1. Each exemplary output queue consists of 8 priority queues.
2. Desire to ensure all 8 queues are flushed, hence need 8 marker frames instead.

3. The CPU inserts 8 proprietary marker frames, one on each of the 8 priority queues.

4. The Egress processing engine (EPE) monitors dequeue of these proprietary marker frames and makes a record, but strip them from the datapath, (preventing the marker frames from being transmitted into network).

5. When the EPE is detected that all the 8 markers had been dequeued, that means traffic from all the 8 queues has been transmitted, the EPE then notifies the processor to insert the real 802.3ad marker PDU frame into the highest priority queue, this marker frame will then be dequeued and transmitted to the remote switch/host.

[0043] C. Conclusion

[0044] Advantages of the invention include the ability to flexibly create a high performance stacked switch with advanced features.

[0045] Having disclosed exemplary embodiments and the best mode, modifications and variations may be made to the disclosed embodiments while remaining within the subject and spirit of the invention as defined by the following claims.